

Analysis Federate

1998

DISTRIBUTION STATEMENT A

Approved for Public Release
Distribution Unlimited

Major William S. Murphy Jr.
U.S. Army TRADOC Analysis Center
PO Box 8692
Monterey, CA 93943-0692
(408) 656-4056
murphyw@mtry.trac.nps.navy.mil

Keywords:

Analysis, advanced distributed simulation (ADS), high level architecture (HLA), analysis federate, composable.

ABSTRACT: *Straight forward implementation of the HLA design principles would require the development of separate, specialized data collection and analysis tools for each federation that uses a different object representation in its Federation Object Model (FOM). The Analysis Federate is a general-purpose data collection and analysis tool that can be used as a composable component in any HLA federation that uses any object representation in its FOM. The reusability of the Analysis Federate across federations requires an extension of federate functionality beyond what is called for in the HLA design principles, which promote the use of a federate as a composable component of one specific federation that uses a single unique FOM. This paper introduces the technologies necessary to extend federate functionality in this manner. These technologies enable the Analysis Federate user to employ tools that automate the subscription, publication, and interpretation of federation data; as well as the ability to automatically generate a unique Simulation Object Model (SOM) for each federation that it is employed with. The Analysis Federate uses the Vision XXI graphical user interface and database. Analysis Federate users must write code to map a federation's objects, attributes, and interactions to the Vision XXI database objects the first time the Analysis Federate is used in every federation that uses a unique FOM. This mapping is necessary to enable the Analysis Federate's analysis and display algorithms to function in a consistent and reliable manner. The Analysis Federate provides the ability to analyze HLA data on a real-time or post-simulation basis.*

1. Introduction

The High Level Architecture (HLA) is an emerging technology that is mandated for use by the United States Department of Defense (DoD) in order to provide a common technical framework to facilitate interoperability between models and simulations. The mandate requires that new DoD simulations being developed must implement the HLA standard, and that legacy simulations must be modified to operate in the HLA framework or be retired. [1] The new HLA communications architecture replaces the protocol based Distributed Interactive Simulation (DIS) standard that was previously used by the DoD. The HLA architecture provides a baseline level of interoperability between federates (simulations) that are operating together in a federation as the components of a distributed simulation exercise. Shifting paradigms from the protocol based DIS to the architecture based HLA introduces new and challenging interoperability problems into DoD distributed simulations that are not satisfied by the HLA baseline. For example, it is difficult to collect and store

the aggregate of the distributed HLA simulation data for analysis or other purposes. These types of interoperability problems must be addressed and solved to ensure that as a minimum distributed HLA simulation users will be able to implement the modeling and analysis capabilities and functions that can currently be implemented under the DIS standard.

Analysis in DIS and HLA distributed simulations is not fundamentally different. The same basic types of algorithms and techniques that are used to analyze data collected during a distributed DIS simulation can also be used to analyze data collected during a distributed HLA simulation. These algorithms and techniques are typically consolidated into an analysis tool that is distributed as a computer program. The only significant difference between analysis in these two simulation environments is the methodology that must be employed to collect and preprocess the simulation data in order to prepare it for use in the appropriate analysis tool. Well established general purpose data collection and preprocessing tools that can be used in any distributed

20000619 092

DTIC QUALITY INSPECTED 4

AA#00-09-2788

DIS simulation are widely available. However, the HLA design principles do not readily facilitate the development or use of similar loosely coupled general-purpose data collection and preprocessing tools that can be used in any HLA federation. Instead, the HLA design principles promote the development of separate, specialized data collection and preprocessing tools (federates) for each federation that uses a different Federation Object Model (FOM). These federates that collect and preprocess data can either be loosely or tightly coupled with the appropriate analysis tool.

The concept of an Analysis Federate was introduced by Jackson in "Exploiting the High Level Architecture for Analysis in Advanced Distributed Simulation." [2] The paper provides a motivation for and description of general data collection techniques in distributed simulations. It also introduces the Study Question Model which should be used prior to a distributed HLA simulation exercise execution to help determine what data should be collected for analysis purposes. However, no practical Analysis Federate implementation details were provided and the level of automation and reusability proposed was minimal.

The development of a general purpose data collection and analysis tool that can be used in any HLA federation requires an innovative solution technique that introduces the concept of federates being reconfigurable and composable components of federations that use different FOMs. This differs from the existing HLA concept of a federate being a composable component of one specific federation and its unique FOM.

This paper provides an integrated technological solution that mitigates the negative impact of the general impediments to interoperability on required HLA federate functionality, and facilitates the development of a general purpose tool that can be used to collect and preprocess data in any federation that is being used to conduct a distributed HLA simulation. This technological solution provides general purpose reusable techniques and procedures that could be used to help automate the creation of federates, eases the programming burden associated with the implementation of HLA interfaces and services in federates, and provides a methodology to facilitate the reuse of federates in federations that use different FOMs. This technological solution focuses on developing a methodology that addresses the integration issues associated with subscribing to objects, attributes, and interactions in an arbitrary FOM for the purposes of building up a change-based historical database of object attribute changes and interactions.

This paper also provides a description of the procedures and techniques that are necessary to develop a general purpose tool that can be used to collect and preprocess data in any federation that is being used to conduct a distributed HLA simulation. It describes the prototype Analysis Federate data collection and analysis tool that was developed to demonstrate this technology. This prototype Analysis Federate provides the capability to collect, process, generate, display, store, access, present, and transfer aggregate FOM data from distributed HLA simulations in order to conduct real-time or post-simulation analysis. The HLA Rules mandate that this prototype tool be implemented as a HLA federate because it will be used to collect, generate, and exchange FOM data during federation execution. The Analysis Federate prototype implements the HLA interfaces, invokes the HLA services, and is adaptable for use with any federation. This will require an innovative approach because it involves the introduction of new functionality into an architecture that was not designed to support it.

The Jackson paper promotes the development of the Analysis Federate SOM from the FOM, other federate SOMs, and from derived data that can be determined by the Analysis Federate during federation execution. [3] The Analysis Federate SOM implemented in the Analysis Federate prototype developed in this research differs from the SOM concept in the Jackson paper. The Analysis Federate prototype uses an automated procedure to generate its own unique SOM for each federation it will join from each federation's FOM.

The benefits to the DoD associated with the demonstrated Analysis Federate functionality include the ability to: answer analysis questions in HLA simulations; provide immediate real time feedback; help exploit situational awareness; improve mission planning and rehearsal; assist in course of action analysis; improve the quality and timeliness of after action reviews; facilitate distance learning; enhance emerging live, virtual, constructive, and synthetic theater of war (STOW) training support systems; and support fielding of the equipment in military units.

2. HLA Interoperability Background

Interoperability in HLA distributed simulations is provided by the HLA rules, interface specification, and object model template (OMT); the three components that define the architecture. Descriptions of these HLA components require the use of the terms federations, federates, and run-time infrastructure (RTI). Federations are the aggregate set of simulations, models, or tools that are used together during a HLA distributed simulation.

Federates are the individual simulations, models, and tools that are members of the federation. The RTI is the distributed operating system for HLA federations. The HLA rules describe the responsibilities of the federation, federates, and RTI. The interface specification is the definition of the interface services between the RTI and the federates. The OMT is the format specification for the object and interaction information contained in the required HLA object models for each federation and federate. These object models are the FOM and the SOM respectively. They facilitate consistent interpretation of data and serve as the HLA interface language. In actuality, FOMs and SOMs are only paper specifications that are used to describe functionality. They are not used online during federation execution in the current HLA design and implementation.

The baseline level of interoperability provided by the HLA architecture between federates in a federation includes the ability to establish a federation of federates, exchange object data between federates, and coordinate federate operations. FOMs facilitate consistent interpretation of exchanged data in a federation by providing a description of objects, attributes, associations, interactions, and level of resolution. Interoperability requirements that go beyond the HLA baseline must be implemented as policy matters for each federation.

The three HLA architecture components implement the HLA design. The premise for this design is that the technical complexity and diverse user needs represented in existing simulations is beyond what is reasonable to handle in a single simulation, and that future technological innovations and simulation uses and requirements could not be predicted. This premise caused systems designers to realize that a composable approach was needed to construct simulation federations. The resulting design principles require modular federates that are designed to participate as composable components in distributed simulation federations that use a single FOM that is structured around the federation's object view of the world. These modular federates have well-defined functionality and interfaces that are separated from the supporting RTI. The HLA design also calls for an object oriented subscription based communications structure in order to reduce the network requirements that are associated with the broadcast based communications structure used in DIS. This subscription based communications structure is implemented by the RTI during federation execution. Federates publish only the data that has been subscribed to by one or more other federates. Federates publish the subscribed data by transmitting it to the RTI executive process which filters

the data and routes it to the appropriate subscribing federates.

These design principles are reflected in the corresponding HLA architecture. This architecture provides the requisite baseline HLA interoperability among federates, but also limits federate interoperability in the following ways. The HLA architecture requires a unique federation specific FOM along with corresponding, compatible, SOMs for each federate in the federation. The requirement for these unique FOMs severely restricts inter-federation interoperability, precludes the use of federates as composable components in federations that do not use the FOM that is compatible with the federate's SOM, and requires all federate and federation developers to document the characteristics of their object representations in the OMT formats. The architecture also requires that all federate designers write computer code to build the functionality required to interface with the RTI in order to participate in a federation and exchange data with other federates. The architecture also limits the transfer of information between federates to the objects and interactions that the individual federates need in order to satisfy their individual requirements. The data transfer requirements and capabilities must be identified as subscription and publication service invocations in the computer code that the federate designers write to interface with the RTI.

These obstacles to federate interoperability in distributed HLA simulations must all be addressed and solved in order to develop a general purpose tool and the supporting methodology that is required to collect, store, and analyze the composite of the distributed simulation data in any HLA federation. The DIS methodology that provides this functionality capitalizes on the broadcast network communications protocols and uses a passive data logger to collect and store all simulation data that is subsequently analyzed after the distributed simulation is complete. The HLA subscription based communications system precludes the use of this DIS passive data logger strategy in distributed HLA simulations. Instead, a new general-purpose methodology must be developed to provide the required data collection and analysis capabilities in distributed HLA simulations. This will require an innovative approach because it involves the introduction of new functionality into an architecture that was not designed to support it.

The data collection and analysis HLA interoperability problem is a special case that is symptomatic of the more general HLA interoperability problems described above. This research will develop an integrated technological solution to these general HLA interoperability problems.

It will demonstrate this integrated solution in a prototype Analysis Federate tool that solves the special case data collection and analysis problem.

3. Systems Analysis and Design

Impediments to interoperability are imposed on the HLA by the basic premise that diverse user needs and simulation complexities make it unreasonable to require the use of a single simulation or object representation. A FOM's object representation facilitates the consistent interpretation of data that can be exchanged in a federation. Each individual federation is required to specify the contents of its FOM. Federates that are capable of joining a federation must publish and implement a SOM object representation that is consistent with the object representation in the federation's FOM. This requirement tends to force the coupling between a federate's SOM and the federation's FOM to be tight. These diverse FOM descriptions combine with the tight coupling between federates and the federation to foster an environment that facilitates the use of federates as composable components within federations that use the same FOM, and as an unavoidable consequence to preclude the use of federates as composable components across federations that use different FOMs. This lack of composability of federates across federations is compounded by the requirement for federate developers to write computer code in order to implement the RTI services. The programming techniques that are taught to potential federate developers call for the federation specific object, attribute, and interaction information to be coded directly into each federate's subscription and publication software. Unfortunately, this technique enforces the tight coupling that exists between federates and their federation.

Accepting the fact that different federations will use different FOMs requires a mitigating solution that does not attempt to impose a universal object model in the HLA architecture. Instead, the composability of federates across federations that use different FOMs must be achieved by developing techniques and tools that can be used to force the coupling between a federate and its federation to be as loose as possible. This will require a paradigm shift in the programming techniques that are used to implement the RTI services in the federates.

3.1 Composability of Federates Across Federations

A federate must be able to adapt its interface functionality in order to be able to provide itself with the capability to subscribe, publish, interpret, and interact with data that is represented in the different FOM object representations in each individual federation that it will

join. A federate's comprehensive description of the permissible objects, attributes, associations, interactions, and level of resolution that it can exchange and interact with in a federation is contained in its SOM. This implies that in order for a federate to be used as a composable component across multiple federations it must have the ability to modify its SOM. It also implies that a federate must have the corresponding ability to map the data formats specified in the modified SOM into the federate's internal database formats so that the data can be used to influence the outcome of the simulation. A federate must publish the modified SOM that it will use prior to joining a specific federation in order to comply with the HLA Rules.

This research developed a methodology that enables a federate to modify its SOM prior to Federation execution as it is migrated between federations that use different FOMs. This methodology requires that the information contained in the FOM and SOM be treated as data. This requires that the FOM's contents must be parsed into computer memory in order to treat it as data. This methodology obtains the FOM information that it parses into computer memory from the two separate computer files that have the "FOM" and "FED" extensions on the file names. The "FOM" file is the FOM that is recorded in the OMT format and is referred to throughout this proposal. The "FED" file, which is referred to as the FED in this proposal, is generated by a software utility program that uses the FOM and other data as inputs. The FED is in essence a reformatted extract of the FOM data that is used by the RTI during federation execution. The FED does not include the data type information that is used to represent the federation's objects, attributes, and interactions in computer memory. However, this data type information is contained in the FOM, which is not used by the RTI during federation execution.

3.2 SOM Generation and Data Mapping

Treating the FOM and SOM as data enables a software utility program to be developed that provides a graphical user interface (GUI) that can be used as a tool to automate the SOM generation process. The following methodology must be implemented in this tool. First the FOM and FED data should be parsed into memory. This parsed data should then be used to populate a GUI that will provide the user with a point and click capability to specify which FOM data elements can potentially be subscribed to and published by the federate. Finally, a SOM that represents the user-specified information should be generated by the GUI tool. This SOM generation tool is not designed to implement the actual subscription and publication requests for a specific

federation execution. Instead, it is intended to produce a SOM that provides a composite description of the federate's ability to interact with other federation members. Care must be taken to ensure that the federate is able to process the object and interaction information that is specified in the SOM in accordance with the HLA Rules. This requires that the user manually produce a file that maps the FOM's object representations into the federate's database formats that represent the federate's internal object representations. This mapping functionality is not fully automated because of the unique object class hierarchies and byte representations defined in each FOM. These factors require that the mapping must be manually tailored for each federation either by mapping object hierarchies, attributes, or applying program logic, depending on the FOM.

The requirement to manually produce a mapping file or mapping logic removes the federation specific object, attribute, and interaction information from the federate's computer code. For simple value to value mappings, the mapping file approach works well. Computer code or scripted logic that is unique for a FOM is appropriate for more complex mappings such as multiple values in one FOM mapping to a single value in another FOM, or value decomposition mappings such as decomposition of a string into multiple values. The requirement to recompile the federate's computer code every time the federation's object representation changes is eliminated if only a simple mapping data file is required. If logic is required, depending on the representation that is used, recompilation may be required.

The use of mapping files or logic significantly eases the programming burden that is placed on the federate developers. These mappings use the federation's data structures as inputs instead of the RTI byte streams that the programmer would have to use without the proposed research contributions. Using the mapping file and mapping logic instead of embedded FOM specific computer code also loosens the coupling between the federate and the federation.

3.3 Automated Subscription and Publication

Treating the FOM as data enables a software utility program to be developed that provides a graphical user interface (GUI) that can be used as a tool to automate the subscription and publication process. This methodology requires the use of the SOM as a mechanism to control the subscription process, and provides runtime validation of SOM to FOM functionality. The following methodology should be implemented in this tool. First the FOM, FED, and SOM data should be parsed into

memory. This parsed data should then be used to populate a GUI that will provide the user with a point and click capability to specify which SOM data elements will be subscribed to and published by the federate. The GUI tool should then be used to enable the federate to join the federation. Finally, the RTI service invocations that are required to subscribe to and publish objects, attributes, and interactions should be transmitted to the RTI by a software routine that is part of the GUI tool. This software routine should loop through the user selected object and interaction information to generate the appropriate data required to invoke the services. It should also process and publication requests from the other federates. After the subscription and publication is complete this tool ceases to have a role in the federate unless there is a requirement to modify the subscription and publication requests or resign from the federation. The RTI buffer information that represents the data that this tool subscribed to will not be processed by this tool. That functionality will be implemented in another tool that is described below.

3.4 Data Marshalling

Treating the FOM as data enables a software utility program to be developed that provides the ability to automatically parse the RTI buffers into FOM data structures during federation execution. This innovation of parsing the RTI buffers enables federates to implement data marshalling capabilities that are currently non-existent in HLA. It enables the federates to check, during federation execution, if RTI packets are the 'right' size as specified by the 'data types' in the FOM. The RTI buffer parsing also enables federates to identify, during federation execution, federates that violate published FOM data structure standards. Conversely, this tool should prevent the federate from publishing anything that is not in the FOM and SOM. The tool that is developed to implement these functions should also provide the ability to transmit federate published values in response to another federate's subscription requests, and to keep track of which objects and attributes are subscribed to by another federate. Additionally, this tool could be used to implement the RTI 'tick' functionality that is used to transfer control between the federate and the RTI.

4. Analysis Tool

The Analysis Federate prototype that was developed to demonstrate this research uses the Vision XXI graphical user interface (GUI) to provide analysts with a tool that allows them to analyze the data that is collected during a distributed HLA simulation. [4] The Vision XXI GUI allows analysts to perform structured queries of the

database in historical or real-time mode. It presents data and analysis results in tables or graphical displays that are tailored by the user, and allows the transfer of data and information to other analysis tools such as spreadsheets, word processors and graphics programs.

5. Fielding and Testing

The Analysis Federate prototype is functional. Its merits are described in a Naval Postgraduate School thesis that performs a comparison study of existing HLA and DIS distributed simulation analysis methodologies. [5] The prototype was also demonstrated at the Military Operations Research Society Symposium (MORSS) in Monterey, California where it was used to perform real time analysis on a distributed HLA simulation. The thesis and the MORSS demonstration both used a HLA federation that included two Janus simulations communicating with each other over a computer network. Janus HLA functionality was established by using two Protocol Data Unit (PDU) Adapter Software Systems (PASS) and two HLA Gateways. The PASS module translated internal Janus data into DIS PDUs. Then, the HLA Gateway translated the DIS PDUs into the data format specified by the HLA federation. The Analysis Federate subscribed to the data in the Gateway's FOM.

The composability of the Analysis Federate across federations was demonstrated when the Analysis Federate was integrated into the Army's Eagle-MODSAF federation. This federation will be used in the Army Experiment V technology demonstrations.

6. Research Contributions

Implementation of the integrated technological solution and corresponding Analysis Federate prototype described in this paper provided several research contributions. These contributions are all essential components of the set of solutions required to mitigate the impediments to interoperability that are imposed on simulation users by the HLA architecture. Each of these contributions were required in order to develop a general purpose data collection and preprocessing tool that can be used in any HLA federation.

The first research contribution improves interoperability in HLA by developing a methodology that makes it feasible to extend the existing concept of federates being composable components within a unique federation that uses a specific FOM, to the new concept of federates being composable components of many federations that use different FOMs. This new concept of federate

composability across federations does not eliminate the requirement for a federate to comply with the HLA rules. Therefore, a unique SOM must be developed for each federation the federate will join. This research contribution will include provisions that enable this SOM generation to be automated.

The second research contribution improves interoperability in HLA by developing a methodology that makes it feasible to eliminate the existing need to write the FOM specific computer code necessary to invoke HLA subscription and publication services from within a federate. This methodology requires the development of an application program tool that is composable and reusable with any FOM. This application program can be used to streamline both the development of new HLA federates, and the conversion of conventional models and simulations into HLA federates by providing the ability to dynamically subscribe and publish without writing code

The third research contribution improves interoperability in HLA by developing a methodology that will make it feasible to provide currently non-existent data marshalling capabilities to federates. This enables federates to: check, during federation execution, if RTI packets are the "right" size as specified by the FOM; and to identify, during federation execution, federates that violate published FOM data structure standards.

The fourth research contribution can be considered as an enabling technology for the first three contributions. The successful development and implementation of the above research contributions all require the development of a method to automatically represent the federation's data structure standards in computer memory during federation execution. This requires an innovative solution approach because the HLA uses byte streams to transfer object and interaction data between federates. The architecture does not provide a mechanism to access data type information during federation execution. However, this can be accomplished by having all components of the federation's object model available at run time.

7. Future Work

The concept of a single Analysis Federate database standard that uses a conceptual model of the mission space is proposed as a future Analysis Federate enhancement. There are three advantages provided by a standard conceptual model. First, the federation's FOM can be mapped to the conceptual model in the Analysis Federate in a relatively straightforward manner using

software tools. Second, an API can be developed for the Analysis Federate database allowing other analysis tools to access the data in a standard way. Finally, the data in the Analysis Federate database can be described in a standard Analysis Federate SOM. This is a basis for forming a standard federation for interfacing analysis tools with exercise federations. This concept would enable many different analysis tools to join a federation with the Analysis Federate using the Analysis Federate SOM (based on the conceptual model) as a FOM. The Analysis Federate would simultaneously participate as a member of the exercise federation that uses the exercise's FOM specific object models. The Analysis Federate would then consolidate the requirements of all of the various analysis tools, subscribe to the exercise FOM data, map the information into the Analysis Federate database using the conceptual model objects, and publish to the exercise federation any derived data that the analysis tools might generate.

8. Conclusion

The tools and technologies described in this paper can be used to improve interoperability in HLA federations by providing the technology necessary to begin to treat federates as composable components across multiple federations. These tools and methodologies can be used to eliminate the need for federate developers to write code to implement RTI services, and the need to modify a federate's local RTI component code in response to FOM changes. The tools can also be used to minimize the work needed to map federation data into a federate's internal data representation by introducing the use of mapping files, logic, or code. It further reduces the work by providing the ability to use the federation's data structures in the mappings instead of the RTI byte streams that are used by programmers who use existing federate development methodologies. Another potential use for these tools is to streamline both the development of new HLA federates, and the conversion of conventional models and simulations into HLA federates.

The ability to treat federates as composable components across federations enabled the development of an Analysis Federate prototype that provides analysts with the ability to collect and analyze data from any federation. This ability to collect data for analysis provides distributed HLA simulation users with the ability to implement the modeling and analysis capabilities and functions that they can currently implement under the DIS standard.

9. References

- [1] Defense Modeling and Simulation Office, US Department of Defense High Level Architecture Interface Specification, Version 1.1., February, 1997.
- [2] Jackson, Leroy and J. Ralph Wood. Exploiting the High Level Architecture for Analysis in Advanced Distributed Simulation. Paper 97S-SIW-122 in *Proceedings of the 1997 Spring Simulation Interoperability Workshop (SIW)*. Simulation Interoperability Standards Organization, Orlando, Florida.
- [3] Ibid.
- [4] Tapestry Solutions, Inc., Vision XXI--An AAR Capability Resulting from a C3 Perspective. San Diego, California. 1998.
- [5] Knight, Steven D., A Comparison of Analysis in DIS and HLA, Naval Postgraduate School Thesis, Monterey, California, June 1998.

Author Biography

WILLIAM S. MURPHY JR. is a Major in the U.S. Army Corps of Engineers with 16 years of commissioned service. He graduated with a B.S. from the United States Military Academy in 1982 and with a M.S. in Mathematics and Computer Sciences from the Colorado School of Mines in 1992. Major Murphy is a licensed Professional Engineer (PE) in the state of Virginia. He is currently an operations research analyst at the U.S. Army Training and Doctrine Command (TRADOC) Analysis Center (TRAC) in Monterey, California. He is also the Chair of the Military Operations Research Society's (MORS) Computing Advances in Military Operations Research working group.